In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown.

1. (Currently Amended) A method, comprising:

examining an instruction stream of a non-executing thread during execution of an executing thread;

identifying an instruction in the instruction stream;

identifying hardware resources associated with the instruction;

determining whether the hardware resource is available to the instruction of the non-executing thread; and

enabling execution of the ~~a~~ non-executing thread if the ~~based at least on whether a~~ hardware resource is ~~or will be~~ available to the ~~an~~ instruction of the non-executing thread.

2. (Original) The method of claim 1, further comprising switching from the execution of a thread executing an instruction with long or potentially long latency, to the execution of the non-executing thread if the hardware resource is or will be available to the instruction of the non-executing thread.

3. (Original) The method of claim 1, further comprising switching from the execution of an executing thread to the execution of the non-executing thread if the hardware resource is or will be available to the instruction of the non-executing thread.

4. (Cancelled)

5. (Cancelled)

6. (Original) The method of claim 1, wherein enabling comprises enabling execution of the non-executing thread based on whether the hardware resource is scoreboarded.

7. (Original) The method of claim 1, wherein enabling comprises enabling execution of the non-executing thread based on whether a register, a queue, a buffer, a functional unit, an input/output (I/O) device, or a memory sub-system device is or will be available to the instruction of the non-executing thread.

8. (Currently Amended) The method of claim 1, further comprising:

~~enabling execution of the non-executing thread if the hardware resource is available to the instruction of the non-executing thread and~~ not enabling execution of the non-executing thread if the hardware resource is unavailable to the instruction of the non-executing thread.

9. (Original) The method of claim 1, further comprising enabling execution of a second non-executing thread if the hardware resource is available to the instruction of the non-executing thread.

10. (Original) The method of claim 1, further comprising switching from executing at least two executing threads to executing the non-executing thread and at least one other non-executing thread if the hardware resource is available to the instruction of the non-executing thread.

11. (Currently Amended) A method, comprising:

determining a number of hardware resources unavailable to a pending thread while executing a first executing thread; and

switching from ~~a~~ the first executing thread to ~~a~~ the first pending thread ~~based at least on the a number of hardware resources unavailable to the first pending thread~~ if the number of unavailable hardware resources to the first pending thread is less than the number of unavailable hardware resources to a second pending thread.

12. (Currently Amended) The method of claim 11 further comprising determining the number of hardware resources unavailable to a second pending thread ~~and wherein switching comprises switching from the first executing thread to the first pending thread if the number of unavailable hardware resources to the first pending thread is less than the number of unavailable hardware resources to the second pending thread~~.

13. (Currently Amended) An article comprising a storage medium having stored thereon instructions, that, when executed by a computing platform, result in:

examining an instruction stream of a non-executing thread during execution of an executing thread;

identifying an instruction in the instruction stream;

identifying hardware resources associated with the instruction;

determining whether the hardware resource is available to the instruction of the non-executing thread; and

enabling execution of the ~~a~~ non-executing thread if the ~~based at least on whether a~~ hardware resource is ~~or will be~~ available to the ~~an~~ instruction of the non-executing thread.

14. (Original) The article of claim 13, wherein the instructions, when executed, further result in: switching from the execution of a long latency executing thread to the execution of the non-executing thread if the hardware resource is or will be available to the instruction of the non-executing thread.

15. (Cancelled)

16. (Currently Amended) An apparatus, comprising:

a hardware resource; and

a thread dispatch circuit to determine if a hardware resource is available to a pending thread while executing an executing thread and switching from the executing thread to the pending thread if the hardware resource is available to the pending thread ~~enable execution of a pending thread based at least on whether the hardware resource is or will be available to an instruction of the non-executing thread~~.

17. (Original) The apparatus of claim 16, wherein the thread dispatch circuit is adapted to examine instructions of multiple threads to determine which threads are executed and which threads remain in a pending state.

18. (Original) The apparatus of claim 16, wherein the hardware resource is a functional unit adapted to perform arithmetic and logic operations or to execute load and store instructions.

19. (Original) The apparatus of claim 16, wherein the hardware resource is a multiplier, adder, a divider, an integer arithmetic logic unit (ALU), a floating point arithmetic logic ûnit

(ALU), a register, a load/store unit, a memory management unit (MMU), a multimedia accelerator, a security coprocessor, or a cryptographic coprocessor.

20. (Original) The apparatus of claim 16, wherein the thread dispatch circuit is adapted to determine which hardware resources are associated with the instructions of multiple threads and to determine if the hardware resources are available to the instructions of the multiple threads.

21. (Original) The apparatus of claim 16, further comprising: an instruction cache to store instructions of multiple threads; an instruction decoder coupled to the thread dispatch; and a register lookup to provide a physical register identification of a register in a register file unit and assign registers to each instruction.

22. (Original) The apparatus of claim 16, wherein the thread dispatch circuit is adapted to switch execution from a running thread to the pending thread.

23. (Original) The apparatus of claim 22, wherein the thread dispatch circuit is adapted to examine instructions of the running thread to determine if a predetermined condition occurs.

24. (Original) The apparatus of claim 22, wherein the thread dispatch circuit is adapted to examine instructions of the running thread to determine if a long latency instruction occurs.

25. (Currently Amended) A system, comprising:

an antenna;

a processor coupled to the antenna, wherein the processor comprises: a hardware resource; and

a thread dispatch circuit to determine if a hardware resource is available to an instruction of a non-executing thread during execution of an executing thread and switching execution from the executing thread to the pending thread if the hardware resource is available ~~enable execution of a pending thread based at least on whether the hardware resource is or will be available to an instruction of the non-executing thread~~.

26. (Original) The system of claim 25, wherein the hardware resource is a functional unit adapted to perform arithmetic and logic operations or to execute load and store instructions.

27. (Original) The system of claim 25, wherein the system comprises a wireless phone, wherein the wireless phone comprises: the antenna; and the processor coupled to the antenna.